This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Previously Presented) A method of locating classes in a class path, the method comprising:

generating a cache of information relating to the classes in the class path;

creating a wrapper for selected elements in the class path to provide a level of indirection from application programming interfaces used by a class locator, the wrapper indirection level providing for different caches to be used for the selected elements;

requesting a search of the class path via the wrapper; and

searching the cache to satisfy the requested search.

2. (Original) A computer readable medium having instructions stored thereon for causing a computer to perform the method of claim 1.

3. (Original) The method of claim 1 wherein the class path comprises multiple elements, each element having multiple classes stored therein.

4. (Original) The method of claim 3 wherein at least one of the elements comprises a ZIP file.

5. (Previously Presented) A method of locating classes in a multi element class path, the method comprising:

generating a search request for desired classes within the multi element class path;

forwarding the search request to a wrapper providing a level of indirection to search the appropriate class path for the search request; and

independently satisfying the request in association with each element in the class path, wherein at least two of the elements have at least two separate caches of information sufficient to satisfy the request for the at least two elements.

6. (Original) A computer readable medium having instructions stored thereon for causing a computer to perform the method of claim 5.

7. (Original) The method of claim 5 wherein at least one of the elements comprises a ZIP file.

8. (Original) The method of claim 5 wherein the classes comprise Java classes.

9. (Original) The method of claim 5 wherein at least one of the elements comprises a Java Package Manager.

10. (Previously Presented) A method of creating caches for selected elements of a class path, the method comprising:

parsing the class path into names of elements;

determining which elements are viable for caching; and

initiating creation of at least two caches for the selected elements and initiating creation of wrappers for each selected elements which is viable, the wrappers providing a level of indirection from application programming interfaces used by a class locator to search the classes.

11. (Original) The method of claim 10 wherein the viability of an element for caching is dependent on the ease of tracking where elements have had changes in them.

12. (Original) The method of claim 10 wherein the viability of an element for caching is determined based on it being a predetermined type.

13. (Original) The method of claim 10 and further comprising checking a registry to see if the element already has a cache associated with it.

14. (Original) The method of claim 13 and further comprising determining if an existing cache is up to date.

15. (Previously Presented) A class path manager comprising:

means for receiving requests to search a multi element class path for classes;

means for transferring such requests through a wrapper providing a level of indirection associated with each element to invoke element specific search methods, the search methods using different caches for different elements.

16. (Original) The class path manager of claim 15, wherein at least one such element specific search method comprises searching a cache associated with such element.

17. (Previously Presented) A class path manager for a multi-element class path, the manager comprising:

means for parsing the multi element class path into names of elements;

means for determining whether each element is a viable cache candidate;

means for creating different caches for such viable candidates; and

means for creating indirection wrappers for each element to map class searches to each element for independent handling.

18. (Original) The class path manager of claim 17 wherein the cache for each viable candidate comprises a name of the class.

19. (Original) The class path manager of claim 17 wherein the elements are selected from the group consisting of directories, ZIP files and Java Package Manager.

20. (Original) The class path manager of claim 19 wherein the directories are not cached.

21. (Original) The class path manager of claim 17 wherein the viability of an element for caching is dependent on the ease of tracking which elements have had changes in them.

22. (Previously Presented) A system for finding classes in a multi element class path, the system comprising:

a class path manager that receives requests for identification or enumeration of classes in the class path;

a cache for each cache viable element of the class path;

a wrapper for each such cache viable element that receives such requests from the class path manager and that provides a transparent level of indirection to services that are specific to such cache viable element.

23. (Previously Presented) A computer readable medium having instructions stored thereon for causing a computer to perform a method of locating classes in a multi element class path, the method comprising:

generating a search request for desired classes within the multi element class path;

forwarding the search request to a wrapper providing a level of indirection to search the appropriate class path for the search request; and

independently satisfying the request in association with each element in the class path, wherein at least two of the elements have at least two separate caches of information sufficient to satisfy the request for the at least two elements, wherein changes to the element result in recreation of the cache.

24. (Original) The computer readable medium of claim 23 and further having instructions for causing the computer to perform:

checking a date/time stamp on the element having the cache of information to determine if the cache is up to date.